Homework 8: From Language to Logic

Course: CS 221 Spring 2019
Name: Bryan Yaggi

In this assignment, you will get some hands-on experience with logic. You'll see how logic can be used to represent the meaning of natural language sentences, and how it can be used to solve puzzles and prove theorems. Most of this assignment will be translating English into logical formulas, but in Problem 4, we will delve into the mechanics of logical inference.

Problem 1: Propositional Logic

(a) coding

(b) coding

(c) coding

Problem 2: First-Order Logic

(a) coding

(b) coding

(c) coding

(d) coding

Problem 3: Liar Puzzle

(a) coding

Problem 4: Logical Inference

Having obtained some intuition on how to construct formulas, we will now perform logical inference to derive new formulas from old ones. Recall that:

- Modus ponens asserts that if we have two formulas, $A \to B$ and A in our knowledge base, then we can derive B.
- Resolution asserts that if we have two formulas, $A \vee B$ and $\neg B \vee C$ in our knowledge base, then we can derive $A \vee C$.
- If $A \wedge B$ is in the knowledge base, then we can derive both A and B.

(a) Some inferences that might look like they're outside the scope of Modus ponens are actually within reach. Suppose the knowledge base contains the following two formulas:

$$KB = \{(A \vee B) \to C, A\}$$

Your task: First, convert the knowledge base into conjunctive normal form (CNF). Then apply Modus ponens to derive C. Please show how your knowledge base changes as you apply derivation rules.

Hint: You may use the fact that $P \to Q$ is equivalent $\neg P \vee Q$.

Remember, this isn't about you as a human being able to arrive at the conclusion, but rather about the rote application of a small set of transformations (which a computer could execute).

$$KB = \{(A \vee B) \rightarrow C, A\}$$
$$KB = \{\neg(A \vee B) \vee C, A\}$$
$$KB = \{(\neg A \wedge \neg B) \vee C, A\}$$
$$KB = \{(\neg A \vee C) \wedge (\neg B \vee C), A\}$$
$$KB = \{\neg A \vee C, \neg B \vee C, A\} \qquad \textit{CNF}$$
$$KB = \{A \rightarrow C, B \rightarrow C, A\}$$
$$\frac{A, A \rightarrow C}{C} \qquad \textit{modus ponens}$$

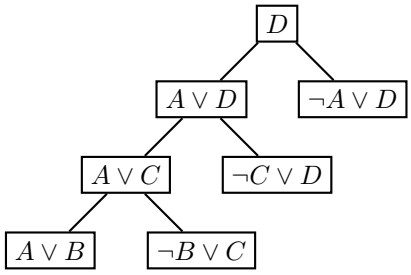
(b) Recall that Modus ponens is not complete, meaning that we can't use it to derive everything that's true. Suppose the knowledge base contains the following formulas:

$$KB = \{A \vee B, B \rightarrow C, (A \vee C) \rightarrow D\}$$

In this example, Modus ponens cannot be used to derive D, even though D is entailed by the knowledge base. However, recall that the resolution rule is complete.

Your task: Convert the knowledge base into CNF and apply the resolution rule repeatedly to derive D.

$$KB = \{A \vee B, B \rightarrow C, (A \vee C) \rightarrow D\}$$
$$KB = \{A \vee B, \neg B \vee C, \neg A \vee D, \neg C \vee D\} \qquad \textit{CNF}$$



Problem 5: Odd and Even Integers

In this problem, we will see how to use logic to automatically prove mathematical theorems. We will focus on encoding the theorem and leave the proving part to the logical inference algorithm. Here is the theorem: If the following constraints hold:

- Each number x has exactly one successor, which is not equal to x.
- Each number is either odd or even, but not both.
- The successor of an even number is odd.
- The successor of an odd number is even.
- For every number x, the successor of x is larger than x.
- Larger is a transitive property: if x is larger than y and y is larger than z, then x is larger than z.

Then we have the following consequence:

- For each number, there is an even number larger than it.

Note: in this problem, "larger than" is just an arbitrary relation, and you should not assume it has any prior meaning. In other words, don't assume things like "a number can't be larger than itself" unless explicitly stated.

(a) coding

(b) Suppose we added another constraint:

- A number is not larger than itself.

Prove that there is no finite, non-empty model for which the resulting set of 7 constraints is consistent. This means that if we try to prove this theorem by model checking only finite models, we will find that it is false, when in fact the theorem is true for a countably infinite model (where the objects in the model are the numbers).